CLOBAS ACADEMIC SOFTWARE

Table of Contents

List of Tables

List of Figures

Abbreviations

SMS: School management system

ASP: Active Server Page

CGAAEB: City Government of Addis Ababa Education Bureau

SQL: Structured Query Language

DBMS: Database management system

IIS: Internet Information Service

HTTP: Hyper Text Transfer Protocol

Abstract

This project work automates school management system. In the system two applications are developed, Windows based (thick client) and Web based (thin client).

The windows application takes most of the activities such as offline student registering, transcript and report card generation and producing the timetable. The web application facilitates attendance recording by the homeroom teachers, to view status of students by their parents and to view reports by kebele and kifle-ketema education bureau officials.

Our solution of the timetable is very simple. In the high school considered for the project there are ten subjects for both grade nine and grade ten. Loads are assigned to each subject teacher and a code is given for each teacher-subject combination. A simple search technique has been used during allocation of each teacher-subject code to a time slot. A database has been used to enforce constraints and to store data.

The prototype has been tested with data from Kokebe Tsebah Secondary School. It has been observed that the system successfully registers students, facilitates attendance recording by the home room teachers and generates various reports such as report card, transcript and a feasible timetable satisfying the constraints (requirements). It has also been shown that the system facilitates to view the status of students by their parents using the Internet or Intranet of the school.

Chapter 1

Introduction

1.1 Background

Education system forms the backbone of every nation. And hence it is important to provide a strong educational foundation to the young generation to ensure the development of open-minded global citizens securing the future for everyone. Advanced technology available today can play a crucial role in streamlining education-related processes to promote solidarity among students, teachers, parents and the school staff.

Education is central to development. It is one of the most powerful instruments for reducing poverty and inequality and lays a foundation for sustained economic growth. With this aim currently our government has given special emphasis to the educational sector and school improvement activities such as continuous professional development for teachers, training and upgrading teachers and capacitating schools with manpower and materials are among the major actions which have been taken in both primary and secondary schools. In order to facilitate and simplify these actions one of the major tool is to have automated school management system.

School Management System(SMS) consists of tasks such as registering students, attendance record keeping to control absentees, producing report cards, producing official transcript, preparing timetable and producing different reports for teachers, parents, officials from kebele or kefle ketema education bureaus and other stakeholders.

Automation is the utilization of technology to replace human with a machine that can perform more quickly and more continuously [2]. By automating SMS documents that took up many large storage rooms can be stored on few disks. Transcript images can be annotated. It reduces the time to retrieve old transcripts from hours to seconds. However, the school system in the government schools of Addis Ababa is not automated and the record officers generate transcripts and reports manually and the school administrators use their experienced knowledge of miss and hit approaches to prepare timetables.

1.2 Statement of the Problem

To help promote students achievement and success, schools must have access to complete, accurate, and timely information about students. One of the benefits of automated SMS is that the student record system will simplify retrieval of required information and is a great instrument for school improvement by taking measures from the information acquired. Despite the use of automated SMS, the government schools in Addis Ababa are using paper based documentation system for performing various tasks and the school administrators apply their knowledge of hit and miss approach in scheduling classes and courses (preparing the timetable) which wastes manpower and much time unnecessarily that does not utilize the current technology.

Transcripts of students are prepared manually by the record officer and teachers. Report cards are produced by the home-room teachers. Attendance of students is recorded by the home-room teachers. In order to control absentees and know the number of days that a student has been absent from the school during the school days the attendance officer has to collect the attendance slips from the corresponding homeroom teachers and compile it which is also a time taking process. In addition to that retrieving records of students who have graduated couple of years ago has been a difficult task and the manual system also has difficulty of producing different reports which are required by the stakeholders such as teachers, administrators or officials from kebele and kifle-ketema.

Teachers may want to associate a student with his parent or emergency persons for disciplinary measures which need searching of the students record in the record office. It has been difficult to search a record from thousands of such records and observed that students can take any person claiming that he/she is their parent or emergency person which creates problem in control of students.

Due to the inefficiency of the current manual system, the need arises to automate SMS in order to efficiently handle students' attendance, to produce transcript, report cards and the various reports satisfying users and customers and to produce timetable which can schedule courses for teachers and classes of students.

1.3 Objective

The general and specific objectives of the project are described below:

1.3.1 General Objective

The general objective of the project is to automate the SMS.

1.3.2 Specific Objectives

In order to attain the general objective, the following list of specific objectives is set:

- To develop an offline registration system,
- To facilitate attendance record keeping,
- To facilitate various report generation,
- To allow teachers, parents, school community and Education bureau officials to view reports on students,
- To produce a timetable

1.4 Organization of the Document

This report document contains seven chapters including this chapter. Chapter two defines and describes concepts with regard to SMS, aiming to give a general view to the reader of the document about tasks or activities which need automation in the school environment. Chapter three presents review of research works on SMS. In chapters four and five, we presented the analysis and design of the developed system respectively. In the remaining chapters, prototype development and conclusion and recommendations are briefly explained.

Chapter 2

Overview of the School Management System

This project emphasizes on school management system in Ethiopian secondary schools. Therefore, we give an overview of the management system of secondary schools in Ethiopia.

2.1 Secondary School in Ethiopia

Secondary education follows eight years of primary education and is for children aged 14 and above.

At the beginning of each academic year which starts in September (Ethiopian New Year), the students get registered and assigned rooms. Each class (section) of students is assigned to a fixed room. Home room teachers are assigned to each class of students. There are two semesters per year. The first semester final examination is usually administered during January, the second semester final examination is administered during the end of June and consequently the results of each class of students is collected, organized, ranked by the corresponding home room teacher and reported to each student. The homeroom teacher also records attendance of each student on each school day which is later organized by the attendance officer. A student who has been absent for more than twenty days is not allowed to take a semester final examination and will be forced to withdraw.

Transcripts are generated by the record officer. A student may request transcript when he/she wants to transfer to other school or when he/she has completed/graduated from the school and needs to join higher education or for some other purpose.

Officials from kebele and kifle-ketema education bureaus want to get statistical reports like number of registered students at the beginning of every year, number of drop outs, and number of passes/failures for each subject at the end of each semester as well as number of passes/failures at a grade level to help them participate in decision making.

2.2 The Timetabling Problem

School timetabling is a major administrative activity in any school. A number of subjects taught by the corresponding teachers are allocated into a number of available classrooms and a number of timeslots, subject to constraints.

The tasks that are considered in constructing the timetable are:

- Assigning periods to classes. There is a need to spread out lessons across the teaching cycle as much as possible, e.g. to avoid having 3 lessons on the same day.
- Some classes need 'double periods' (preferably 2 consecutive periods). This happens currently for Mathematics and English since each of the subjects have 6 lessons per week (for five days) and therefore on one of the days these subjects should have two lessons for each class of students.

Some combinations of assignments lead to acceptable timetables, others do not. Such restrictions follow from conditions imposed by classes (rooms), students or teachers. We distinguish two types of conditions: conditions that must be met (*requirements*) and conditions that should be fulfilled as well as possible (*desires*) [5]. A class of students has a fixed room throughout the academic year and therefore we use class and room alternatively. Here a room is mentioned only when there are students in it.

The time tabling problem is said to be feasible if and only if it satisfies the following constraints (*requirements*):

- Every teacher and every class must be present in the timetable in a predefined number of periods;

- There can not be more than one teacher in the same class at the same period;
- No teacher can be assigned to more than one class at the same time;
- There can be no "uncovered periods" (that is, periods when no teacher has been assigned to a class).

Violation of the constraints leads to an infeasible solution which usually happens in the manual preparation of the timetable. Therefore, these constraints have to be satisfied in order to get a *feasible solution.*

2.2.1 Time Slot Assignment

The school timetabling is a weekly scheduling for all the classes of a school, avoiding teachers meeting two classes at the same time, and vice versa. This means that an event may be placed in the timetable only in such a way that it does not violate constraints.

Figure 2.1 shows the concepts of timetable construction at schools. Lessons in a subject are taught by a teacher to a corresponding class of students and the timetabling problem is a problem of allocating resources, i.e. assigning to teachers and class of students, time slots and lessons [3, 4]. A time slot is a period and a lesson is an event associating a teacher, a subject and a class of students with in a time slot.

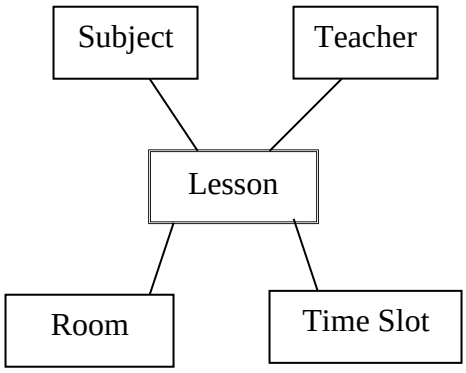


Figure 2.1 Concept of timetabling construction at schools

Timetabling is based on a rectangular time grid that divides the planning period into disjoint time intervals of equal duration (42 minutes in this case) which are called time slots or simply periods. Table 2.1 shows a typical time grid with one day (Monday) for all possible classrooms ($\mathbf{r}_i$) and seven time slots a day.

Table 2.1 Timetable for one day (Monday) of the five days

.		**r1**	**r2**	**r5**	**…..**	**rn**
Monday	p1	e1	e22	e6	….	
	p2	e3	e5	e7	….	**e9**
	p3	e8	e10	e15	….	**e18**
	p4	e12	e34	e20	….	**e11**
	p5	e32	e27	e24	….	**e17**
	p6	e26	e19	e25		**e23**
	p7	**e28**	**e21**	**e29**	**…**	**e35**

If there are m classes i=1,…,m, n teachers j=1,…,n, and T timeslots t=0,…,T-1., then the total number ($\mathbf{r}_{ij}$) of lectures which is commonly called load of the teacher is known in advance. Each lecture takes one time slot.

Currently in the government high schools in Ethiopia, there are five school days from Monday to Friday and there are seven periods per each school day. A time table for each class or teacher could be prepared on a grid as shown in table 2.2. A cell in the grid represents teacher and subject combination in the case of class/section time grid and the class that a teacher teaches in the case of a teacher time grid.

Table 2.2 Typical time grid with five days and seven time slots a day

	Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
p1					
p2					
p3					
p4					
p5					
p6					
p7					

The lectures are activities and teachers and classes are resources. These resources are not available at certain time periods. A lecture can be given in period t only if the corresponding class and teacher are available in t [1]. All lectures have equal processing length, i.e. the length of the period and have to be scheduled without preemption.

To automate the school activities some literature reviews have been done. The literature reviews is discussed in chapter 3.

Chapter 3

Literature Review

Automated SMS plays a great role in simplifying the job of employees at the school and satisfying the need of customers and stakeholders of the school. Even though no documentation is found in Ethiopia to be reviewed, products have been observed at some schools to help understand the problem of managing schools and handling school data. This chapter reviews these products.

3.1 Observed Products

In the year 2003 City Government of Addis Ababa Education Bureau (CGAAEB) was very much interested to have automated school management system to get uniform and quick access to the students' data for administrative purpose on promoting the students' achievement and related issues. The bureau has selected Wundrad Preparatory School for pilot test. At the time the school principals together with officials from CGAAEB signed a contractual agreement with some software developer company. The developers installed their first version of the product which can register a student offline and generate official transcript with some level of difficulty. As the system is not fully automated, it does not support management of attendance, does not support generating report cards and other important functions such as generating school timetable and a web based report for parents. Due to the lack of follow up by the government officials at CGAAEB, the company was unable to complete the project. The school currently is unable to use the partially developed system because of lack of trained person and lack of hardware and software maintenance.

Another product that is in use is transcript generator system. The transcript generator system at Menelik II Preparatory School generates official transcript of students. In order to generate transcript the record officer enters the student information along with the grade marks for the grades completed per year and per semester. Then the system generates the required official transcript. Currently the school is using the system to generate official transcript even though the data entry format has unnecessarily many fields which are not applicable for the record office but can be used for continuous assessment by the course teacher.

3.2 Manual Timetabling

Manual timetables are prepared by dedicated teachers. In manual timetabling, it is common to proceed in an iterative fashion where each iteration selects and schedules a lesson [3]. Scheduling a lesson requires to choose a classroom (fixed for each section of students) and a time slot such that the commitment to the choice will not violate any constraint.

In school timetabling, we are required to schedule a given set of meetings such that the resulting timetables are feasible and acceptable to all people involved. Humans are able to prepare the timetable using some hit/miss approach. So it is possible to automate the timetable based on a simulation of the human way of solving the problem. Such techniques, that we call direct heuristics, were based on a successive augmentation. That is, a partial timetable is extended, lecture by lecture, until all lectures have been scheduled. The underlying idea of all approaches is to schedule the most constrained lecture first.

Usually some responsible teachers are assigned to schedule subjects and teachers. The number of teachers available per each subject is predefined and the load that each teacher has is calculated. With these data the timetable constructor assigns each teacher-subject association to the appropriate classes with the available time slots.

The manual solution of the timetabling problem usually requires many person-days of work. In addition, the solution obtained may be unsatisfactory. The lessons should be fairly distributed satisfying the identified constraints.

3.3 Drawbacks of the Reviewed Systems

The reviews described have the following problems:

- Generate official transcript with some level of difficulty,
- Do not sufficiently produce the required reports to allow parents to view status of their children and reports for officials of kebele and kifle-ketema to help them participate in decision making,

- Do not generate timetable for the schools
- Do not facilitate attendance record keeping by the homeroom teachers

This project work tries to fill the gap by automating the various activities at schools. It tries to satisfy customers need and simplify the works of administrators, record officer and teachers. With an automated school management system parents can easily interact with the school community to follow up their children's achievement and play their role in the school development processes.

Chapter 4

System Analysis

In this chapter the functional and non-functional requirements of the system are described and modeled using UML models.

4.1 Functional Requirements

The functional requirements of the system are:

- register a student,
- record attendance of students,
- generate various reports,
- generate timetable.

4.2 Non Functional Requirements

Security requirements are important factors in this system as classified data will be stored in the database. User validation will be done during login to insure that the user is valid and that the user only has access to his or her permission data. General users will only have access through the user interface.

The system will have consistent interface formats and button sets for all forms in the application, will have a form based interface for all data entry and viewing formats, and will generate reports that are formatted in a table and that should look like the existing manual report formats for user friendliness.

The system will be easily maintained by the developer or other authorized trained person and it shall respond as fast as possible in generating report and producing the timetable.

4.3 Analysis Model

To produce a model of the system which is correct, complete and consistent we need to construct the analysis model which focuses on structuring and formalizing the requirements of the system. Analysis model contains three models: functional, object and dynamic models. The functional model can be described by use case diagrams. Class diagrams describe the object model. Dynamic model can also be described in terms of sequence, state chart and activity diagrams. For the purpose of this project we have described the analysis model in terms of the functional model and dynamic models using use case and sequence diagrams.

4.3.1 Use case Diagram

Use cases of the system are identified to be “RegisterStudent”, “RecordAttendance”, “GenerateTranscript”, “GenerateReportCard”, “ViewReport” and “ProduceTimetable”. The diagram depicted in Figure 4.1 shows the use case diagram of the system.

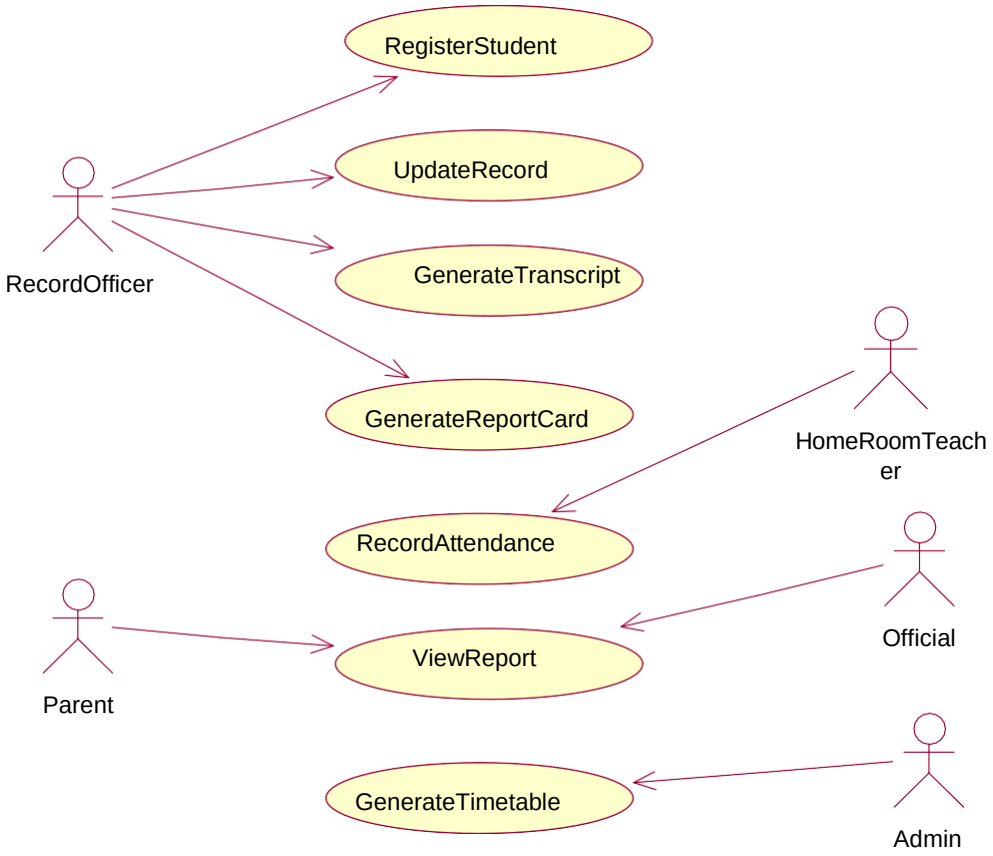


Figure 4.1 Use Case Diagram of the SMS

4.3.2 Actor Description

Name: RecordOfficer

Description: A RecordOfficer is a person who registers a student, input, update student data and produce transcript and report card.

Name: HomeRoomTeacher

Description: A HomeRoomTeacher is a teacher assigned by the school director to each class of students to follow the students closely. He/She has the responsibility of recording attendance of students and submitting.

Name: Parent

Description: A Parent is a person who is registered as parent of the student and responsible to follow the student in close contact with the school. He/She can view the status of the student such as attendance and result/performance of the student online.

Name: Official

Description: An Official is a person who is registered as a legal concerned official from kebele or kifle-ketema to get reports on students of the school. He/She will be viewing the report online.

Name: Admin

Description: Admin is a person who is responsible to produce the timetable for each teacher and classroom by providing the necessary parameters.

4.3.3 Use Case Description

- **Name**: RegisterStudent

Actors: RecordOfficer

Description: To register some one as a student of the school

Precondition: A student has to be eligible (has to be from the pre-specified junior schools that the school will accept)

Flow of Event:

(1) student wants to be registered as a student of the school
(2) Record officer verifies that the student is eligible
(3) Registration form will be given to the student
(4) The student completes the registration form that contains student's full name, address, parent name, emergency person names and addresses and other detail information.
(5) RecordOfficer of the school checks whether the contents of the registration form is properly completed
(6) RecordOfficer fills and submits the form to the system
(7) System registers
(8) Use case ends

Post Condition: Student Registered

- **Name**: RecordAttendance

Actors: HomeRoomTeacher

Description: To record attendance of students in each school day

Precondition: A home room teacher must login as the home room teacher of the class to record attendance

Flow of Event:

(1) A home room teacher wants to record absentees from the class
(2) The home room teacher fills in the attendance slip in the class room
(3) Having the attendance slip the home room teacher logs in to record
(4) HomeRoomTeacher records absentees and submits
(5) System acknowledges
(6) Use case ends

Alternative Flow of Events

Alternative flow A: User is not a home room teacher of the class

A3. User can't record attendance for the required class of students

A4. Use case ends

Name: GenerateReportCard

Actors: RecordOfficer

Description: To produce a report card for students per semester

Precondition: A student must have complete grade marks in all subjects of the semester

Flow of Events:

(1) The record officer logs in and selects the class/section to which the student belongs
(2) The record officer searches the student from the class/section based on the search criteria defined
(3) The system processes the report card
(4) System displays and print the result

(5) Use case ends

Alternative Flow of Events

Alternative flow A: The user logged in is not the record officer

A1. User can not generate report card

A2. Use case ends

Alternative flow B: The student is incomplete at least in one subject

B3. The system can not generate the report card.

B4. Use case ends

Name: GenerateTranscript

Actors: RecordOfficer

Description: To produce transcript based on the request of a student

Precondition: A student must have completed at least a semester to have grade report

Flow of Event:

(1) A student wants to get transcript
(2) The record officer logs in and searches the students record from the database based on the search criteria
(3) The system processes the transcript
(4) System displays and print the result
(5) Use case ends

Alternative Flow of Events

Alternative flow A: The student is incomplete at least in one subject

A3. The system can not generate transcript

A4. Use case ends

- **Name**: GenerateTimetable

Actors: Admin

Description: To generate timetable for teachers and classes

Precondition: There should be subject teachers and classes for which the timetable is to be produced

Flow of Event:

(1) Admin wants to generate timetable
(2) Admin logs in
(3) Admin registers teachers
(4) Admin assign classes for each subject teacher by filling a form and then submits it
(5) The system generates the timetable
(6) System displays the result
(7) Use case ends

Alternative Flow of Events

Alternative flow A: No teacher is registered to be assigned to classes

A5. The system can not generate timetable

A6. Use case ends

Name: ViewReport

Actors: Parent

Description: To view the status of a student

Precondition: A parent must be recorded in the system as parent of the student

Flow of Event:

(1) A parent wants to view status of his/her student
(2) Parent logs in to the system supplying user name and password
(3) System displays a form for search criteria for a student
(4) Parent fills in the form and submit
(5) Parent selects to view the required report
(6) System displays the appropriate report
(7) Use case ends

Alternative Flow of Events

Alternative flow A: Parent is not recorded as parent of a student

A2. The parent can not login

A3. Use case ends

Alternative flow B: parent is unable to give appropriate name and Id number of a student

B5. System can not display the student's information

B6. Use case ends

4.3.4 Sequence Diagrams

Sequence diagrams show the interaction between participating objects in a given use case. They are helpful to identify the missing objects that are not identified in the analysis object model. To see the interaction between objects, the following describe the sequence diagram of each identified use cases.

In Figure 4.2 below, once the user has activated the registeration module by interacting with the boundary object "*NewRegisterationButton*" button, the control object named *"RegisterationControl"* manages the activities involved in "registerStudent" use case. First the "Registeration*Control*" creates registeration form which will be filled by the secretary and submitted. The registration control sends the record to a persistent storage.

The sequence diagrams for "RecordAttendance", "GenerateTranscript", "ViewReport" and "GenerateTimetable" use cases are shown in figures 4.3, 4.4, 4.5 and 4.6 respectively.

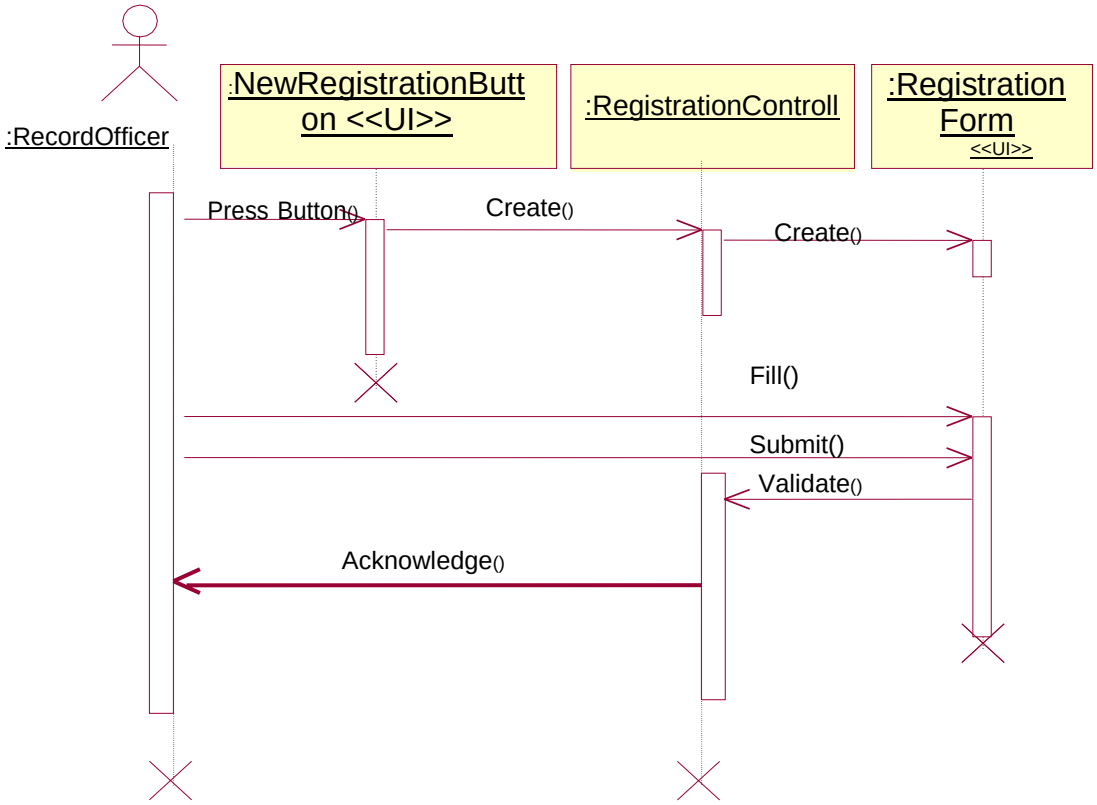


Figure 4.2 Sequence Diagram for Student Registration

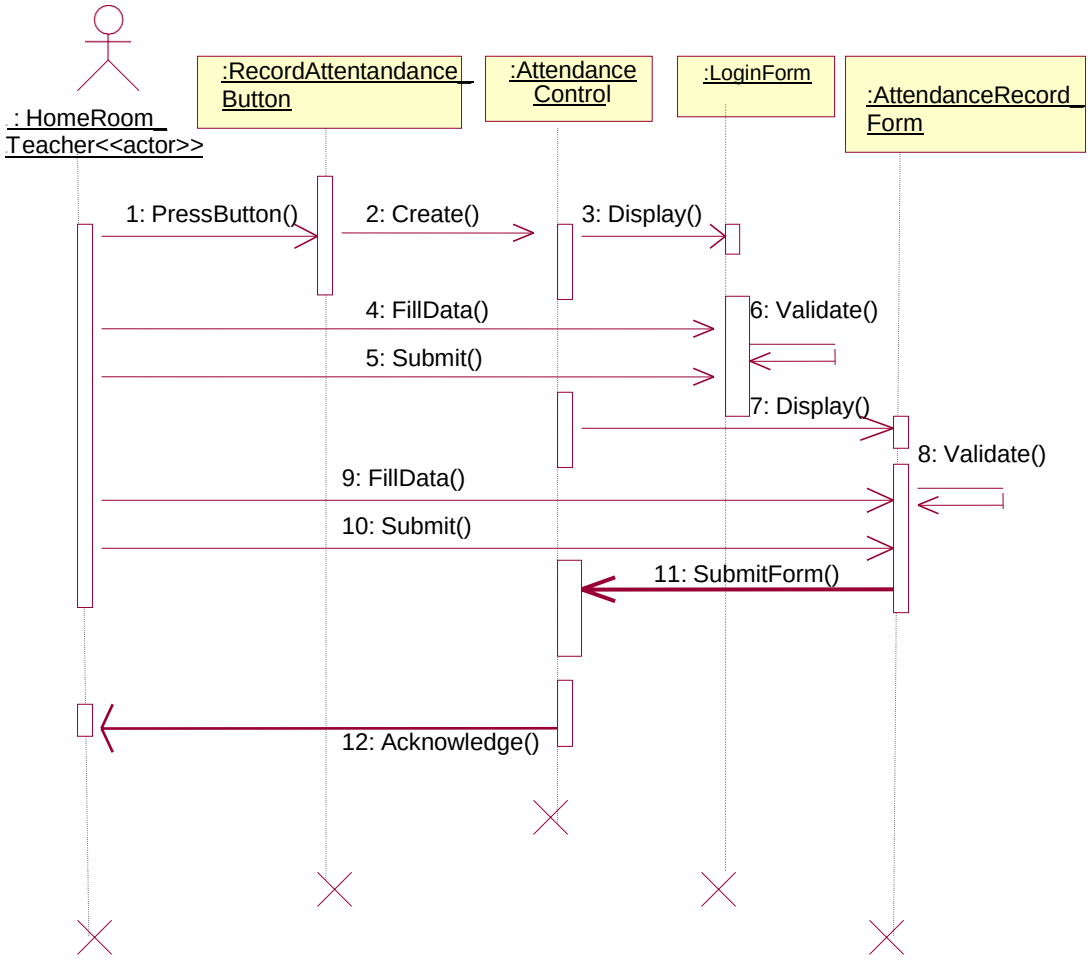


Figure 4.3 Sequence Diagram for Recording Attendance

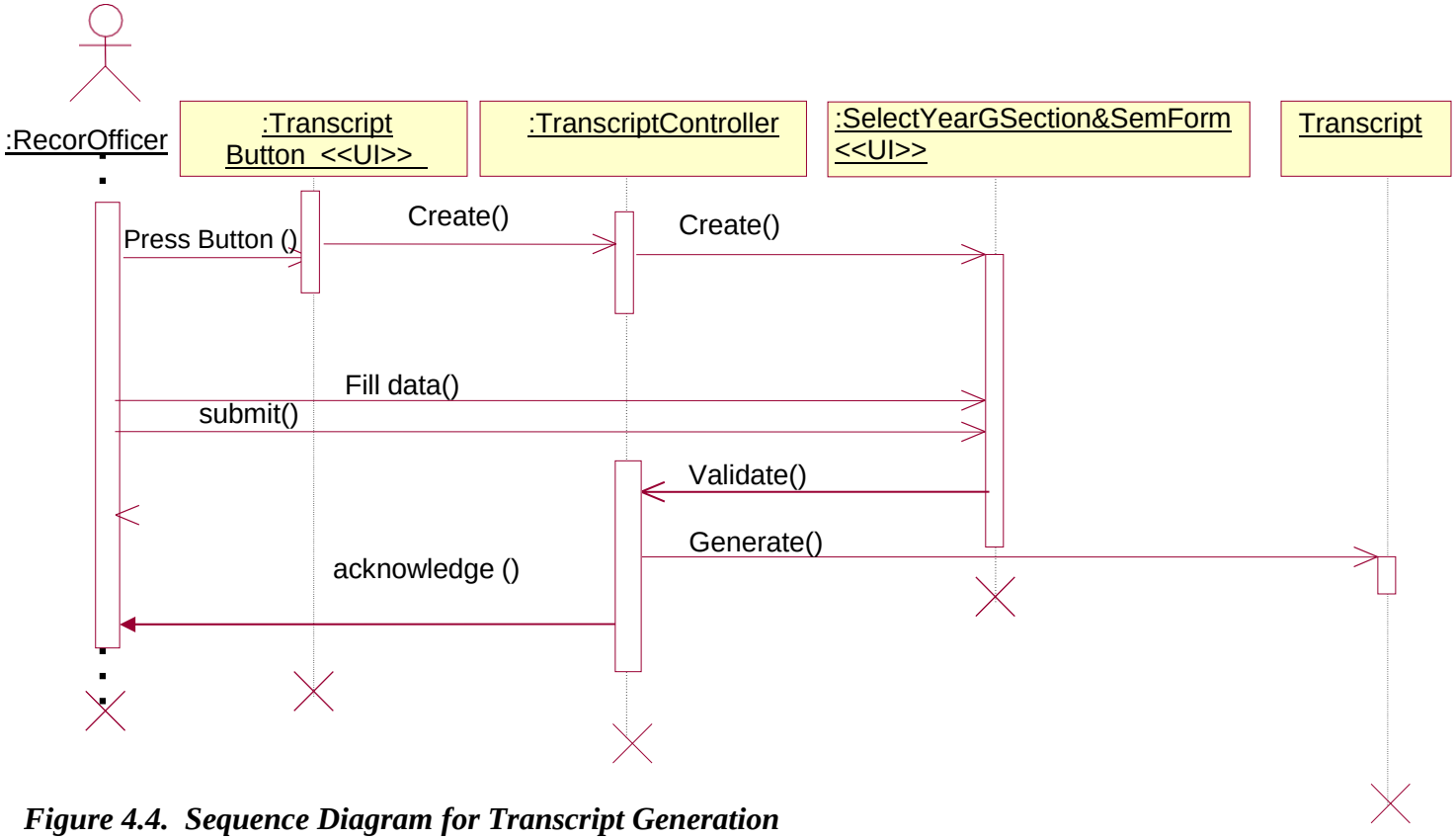


Figure 4.4. Sequence Diagram for Transcript Generation

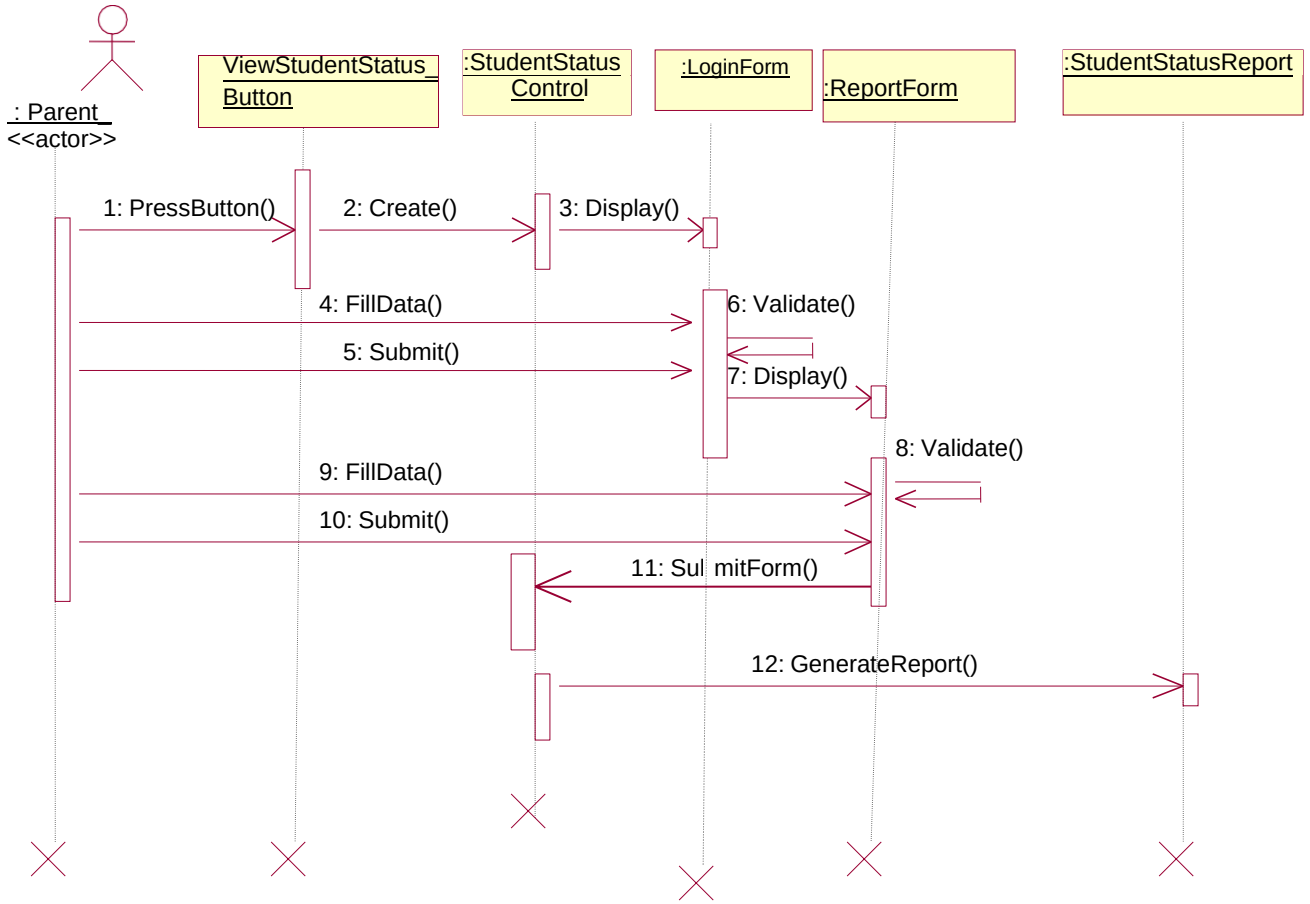


Figure 4.5 Sequence Diagram for viewing student status by the parent

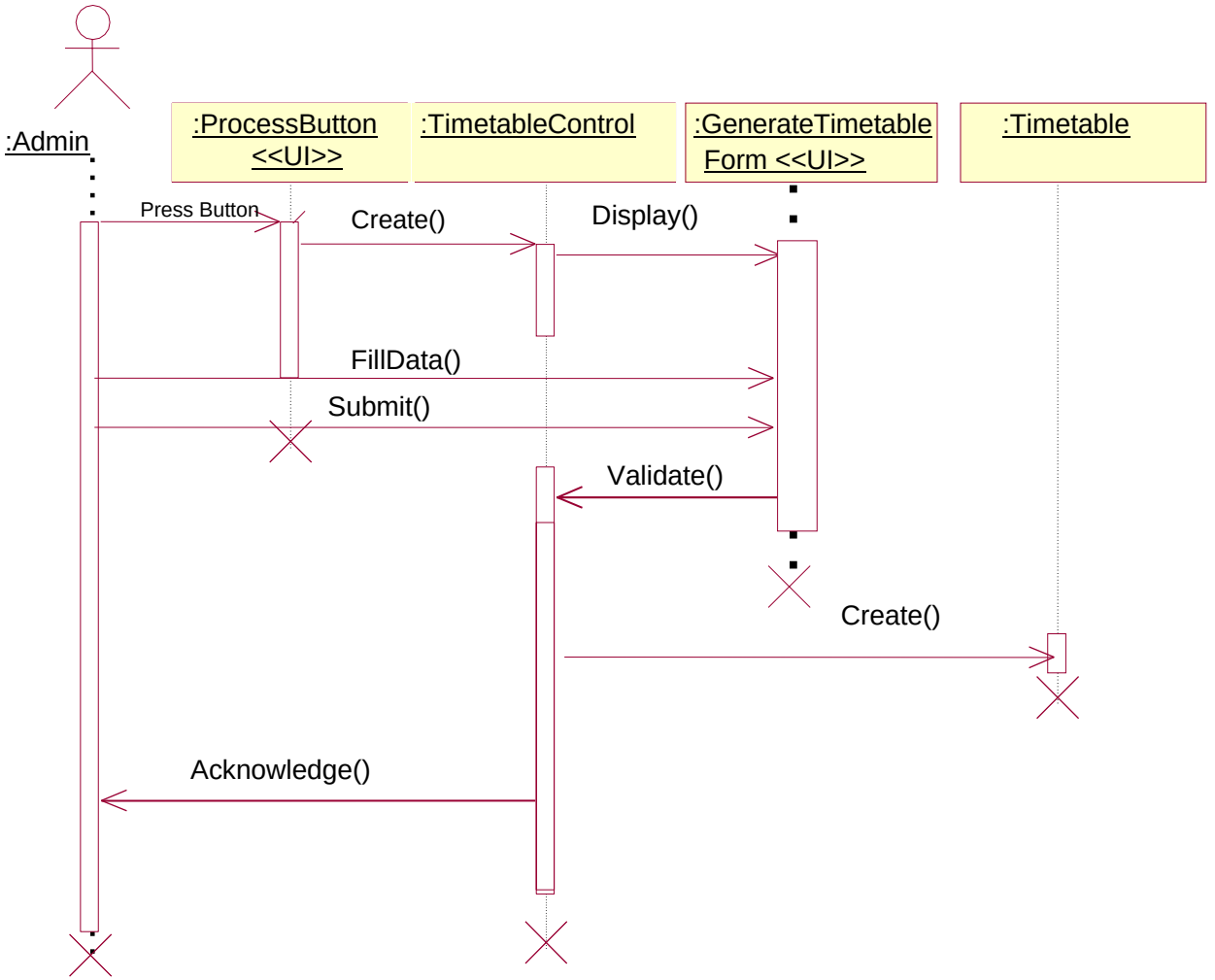


Figure 4.6 Sequence Diagram for generating Timetable

Chapter 5

System Design

In the previous chapter we have identified the functional and non-functional requirements of the system and produced the analysis model. The following are discussed in this chapter: design goals, system architecture, system decomposition, deployment and database design.

5.1 Design Goals

Design goals describe the qualities of the system that developers should optimize. Such goals are normally derived from the non-functional requirements of the system.

Design goals are grouped into five categories. These are

- Performance
- Dependability
- Maintenance
- End User Criteria

5.1.1 Performance Criteria

The part of the system to be used for the record office should have a fast response time (real time) with maximum throughput. Furthermore, the system should not be taking up too much space in memory. The record officer has chosen fast response time over throughput and hence the system should try to be more interactive. In the case of the timetabling subsystem, the system should be more reliable in order to satisfy the constraints than fast response time.

5.1.2 Dependability

The school needs the system to be highly dependable as it is expected to be used by non-IT professionals. The system should be robust and fault tolerant. Furthermore, as the system is handling sensitive data of the school, high emphasis should be given with regards to security, as there are subsystems to be accessed through web.

5.1.3 Maintenance

The system should be easily extensible to add new functionalities at a later stage. It should also be easily modifiable to make changes to the features and functionalities.

5.1.4 End User Criteria

Usability: Usability is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. From the end users' perspective the system should be designed in such a way that it is easy to learn and use, efficient and having few errors if any.

Trade-off is inevitable in trying to achieve a particular design goal. One best case is the issue of security versus response time. Checking User-Id and Password before a member can enter to the SMS creates response time problem/overhead. The other case is the issue of response time versus quality. There is some amount of time taken by the system to generate the timetable. So the user has to wait a little after telling the system to generate the timetable and getting the result to get a quality timetable.

5.2 Architecture of the System

The proposed system is expected to replace the existing manual system by an automated system in all facets. It is mainly based on the system Analysis document (chapter 4).

The architecture used for the system is a 3 tier Client/Server Architecture where a client can use Internet browsers to access the online report provided by the system within the local area network of the school or any where using the Internet. Figure 5.1 shows the architecture of the proposed system.

The data tier maintains the applications data such as student data, teacher data, timetable data etc. It stores these data in a relational database management system (RDBMS).

The middle tier (web/application server) implements the business logic, controller logic and presentation logic to control the interaction between the application's clients and data. The controller logic processes client requests such as requests to view student's result, to record attendance or to retrieve data from the database. Business rules enforced by the business logic dictate how clients can and cannot access application data and how applications process data.

A web server is a program that runs on a network server (computer) to respond to HTTP requests. The most commonly used web servers are Internet Information Server (IIS) and Apache. The web server used in this system is IIS. HTTP is used to transfer data across an Intranet or the Internet. It is the standard protocol for moving data across the internet.

The client tier is the applications user interface containing data entry forms and client side applications. It displays data to the user. Users interact directly with the application through user interface. The client tier interacts with the web/application server to make requests and to retrieve data from the database. It then displays to the user the data retrieved from the server.

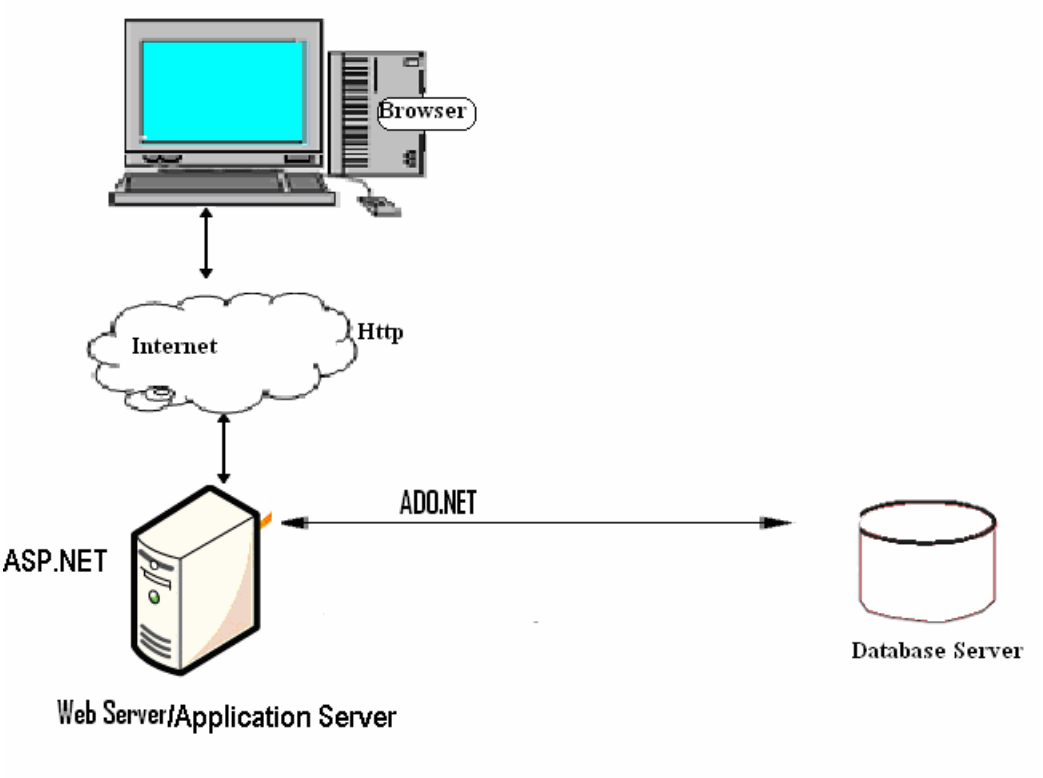


Figure 5.1 Architecture of the System

5.3 Subsystem Decomposition

Subsystem decompositions will help reduce the complexity of the system. The subsystems can be considered as packages holding related classes/objects. The SMS under consideration is decomposed into subsystems as shown in Figure 5.2. These subsystems are further decomposed into other subsystems. The major subsystems identified are “StudentRegistration”, “Login”, “Attendance”, “ReportCard”, “Transcript”, “Timetable” and “Report” subsystems.

Users are classified in to roles. The “Login” subsystem authenticates a user to grant access based on the role of the user. The “StudentRegistration” subsystem registers a

student offline. It allows recording the detail information of the student including parental and emergency person.

“Transcript” and “ReportCard” subsystems are used to generate transcript and report card respectively. The “Timetable” subsystem generates a timetable, which involves allocating a time slot to a subject teacher for a class of students.

The “Attendance” subsystem facilitates recording absent students on the school day by the homeroom teacher to control absentees and to report to parents and the administrator to take corrective measures. The “Report” subsystem generates reports to parents, officials from kebeles and kifle-ketema and teachers in order to facilitate viewing students’ status and course achievement online.

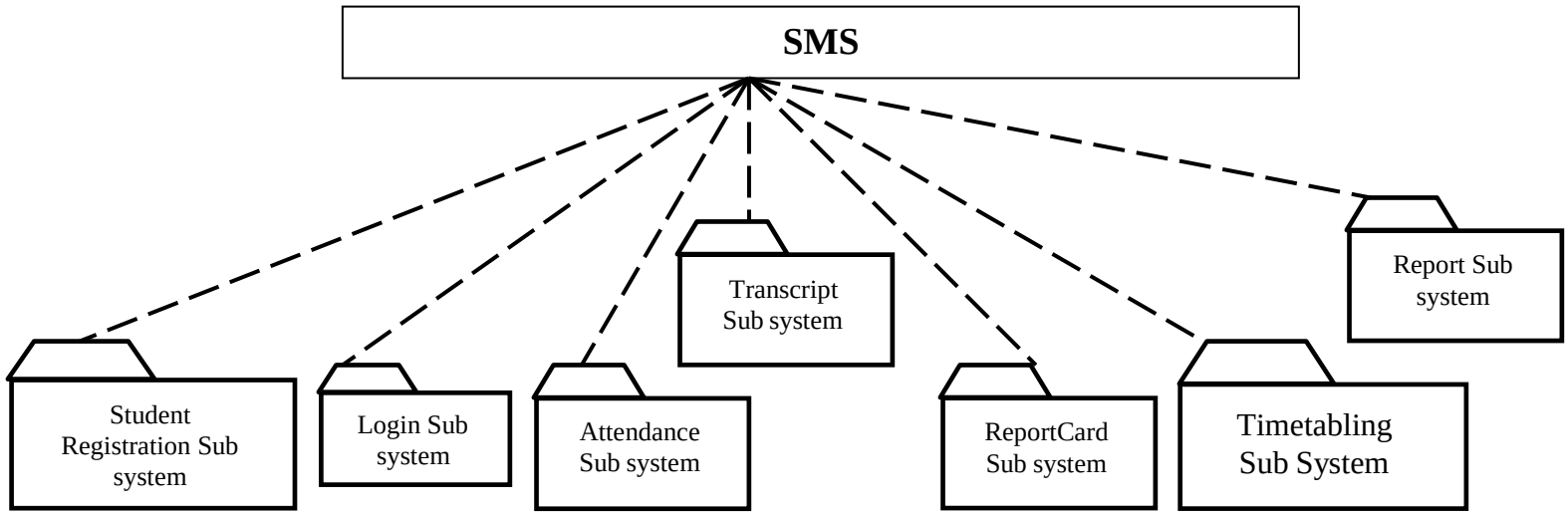


Figure 5.2 Layered Representation of the System

Further decomposition of some of the subsystems is shown in Figure 5.3.

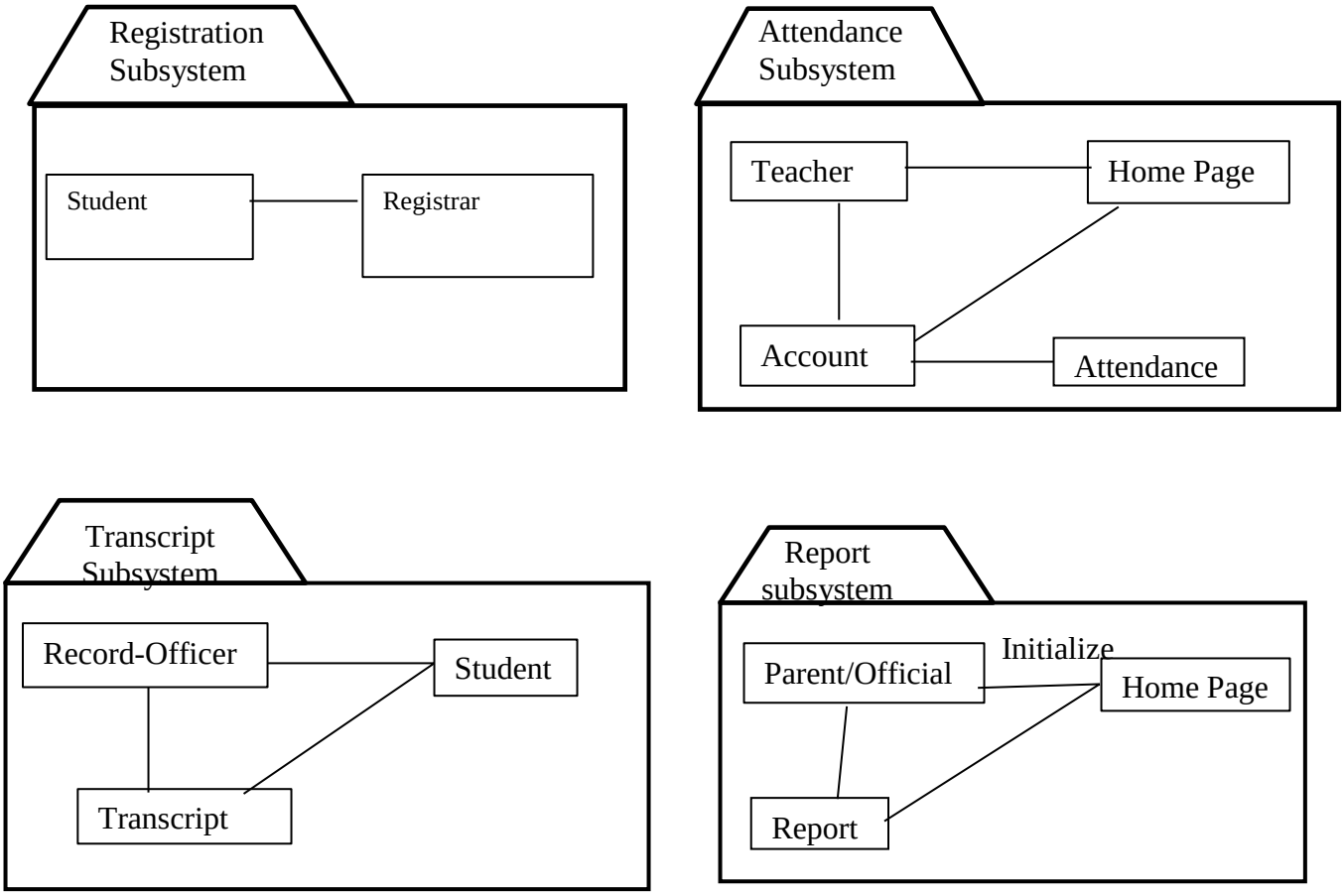


Figure 5.3 Subsystem Decomposition Diagram

5.4 Hardware/Software Mapping

One of the major tasks in system design deals with hardware/software mapping which deals with which components would be part in which hardware and so on. The SMS is a broad system that performs many functions as described in chapter 4. It consists of web based system used by homeroom teachers to record attendance. The web based system also assists parents and officials to get or view status and report on students' achievement and progress. The system assists the record officer to generate transcript and report cards. So the web based part is expected to run on

a networked environment on different Operating System platforms. The client/server architecture of the system enables different clients to connect to the server remotely through Internet connection.

The system has two nodes such as the ***Web server*** and ***Clients.*** These nodes are shown as UML Deployment diagrams in Figure 5.4. The nodes can represent specific instances (workstations) or a class of computers (web server), which is a virtual machine. The applications of the system will run on the web server connected to the database server by ado.net

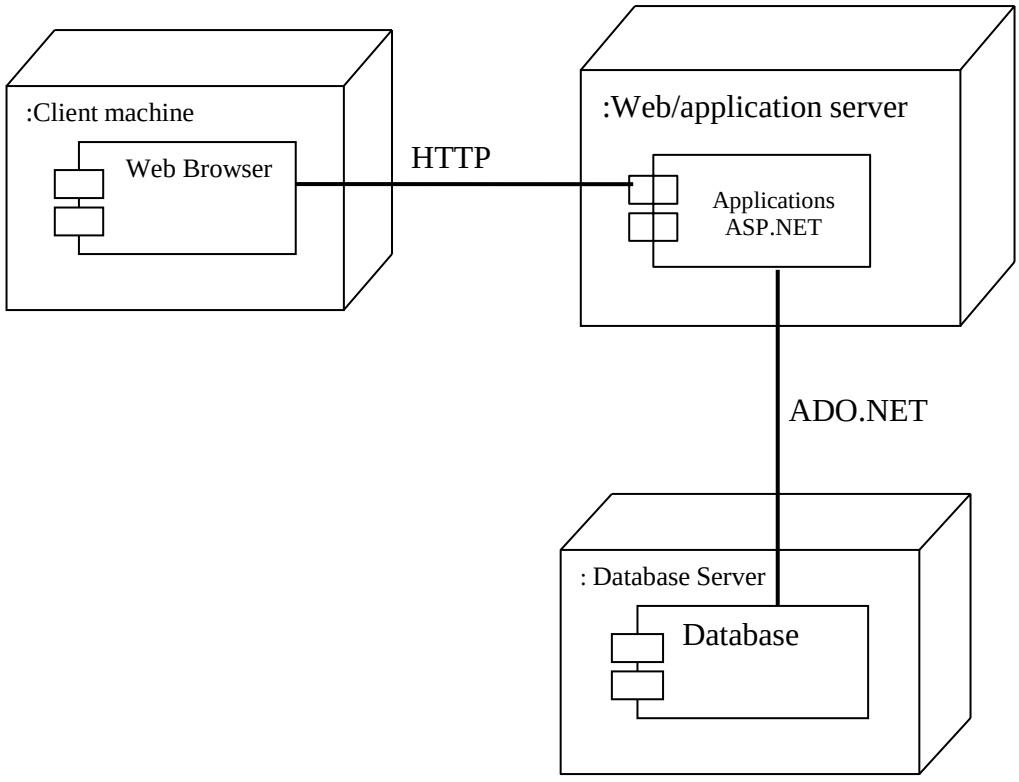


Fig 5.4 Deployment Diagram of the System.

The system has two applications to be developed on the same database, Windows and Web applications. When dealing with windows applications, there are compiled program that must be distributed to the users desktop before they can use it. Depending on the application, there may also be one or more supporting DLLs or other executables such as Crystal Reports [6]. While in thin-client applications (Web applications) there is typically no program or DLL to be distributed.

Users merely need to start their browsers and enter the URL of the application Web site. The server hosting the Web site is responsible for allocating all the resources the Web application requires.

5.5 Persistent Data Management

Persistent data management deals with how the persistent data (file, database, etc) are stored and managed and it outlives a single execution of the system. Information related to student basic information, student's attendance and grade mark, the timetable produced and other related information are persistent data and hence stored on a database management system. This allows all the programs that operate on the SMS data to do consistently. Moreover, storing data in a database enables the system to perform complex queries on a large data set

The schools register students every year in thousands per grade level. For complex queries over attributes and large dataset Microsoft SQL Server is implemented, which is a Relational Database Management System.

5.5.1 Mapping

In order to store information persistently we map objects into tables and the attributes into fields to the specific table based on the objects found on the system. Therefore, we identified the major tables that will be implemented on the selected DBMS. For this reason, some of the mapping of objects to tables is displayed as in Fig 5.5.

5.5.2 Relationships among Tables

This part is to describe and show the necessary relationships among the tables, which are selected to store the data persistently in the system. Generally there are three types of relationships in a relational database system. These are one-to-one, one-to-many and many-to-many relationships. The system under consideration has one-to-many and many-to-many relationships.

Student and AcademicSubject tables have many-to-many relationships. One of the aims in a database system is to reduce redundancy and for that purpose many-to-many relationship has to

be reduced to one-to-one relationship. The Student and StudMarks and the AcademicSubject and StudMarks have one-to-many relationship by using the StudMarks table as the associate table. The relationship of the remaining tables and the ones described here are descriptively shown in Fig. 5.6.

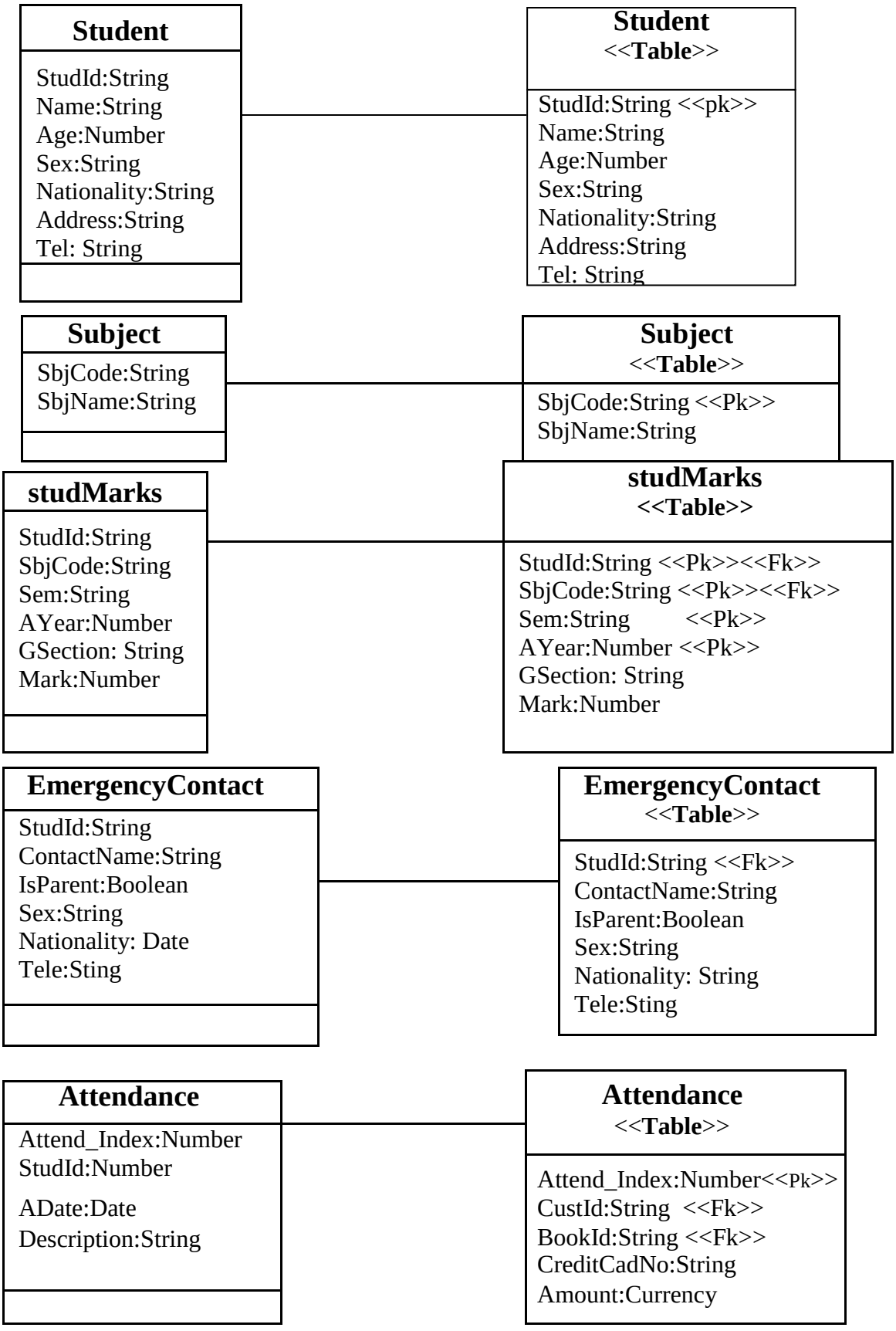


Figure 5.5 Mapping Objects into Tables

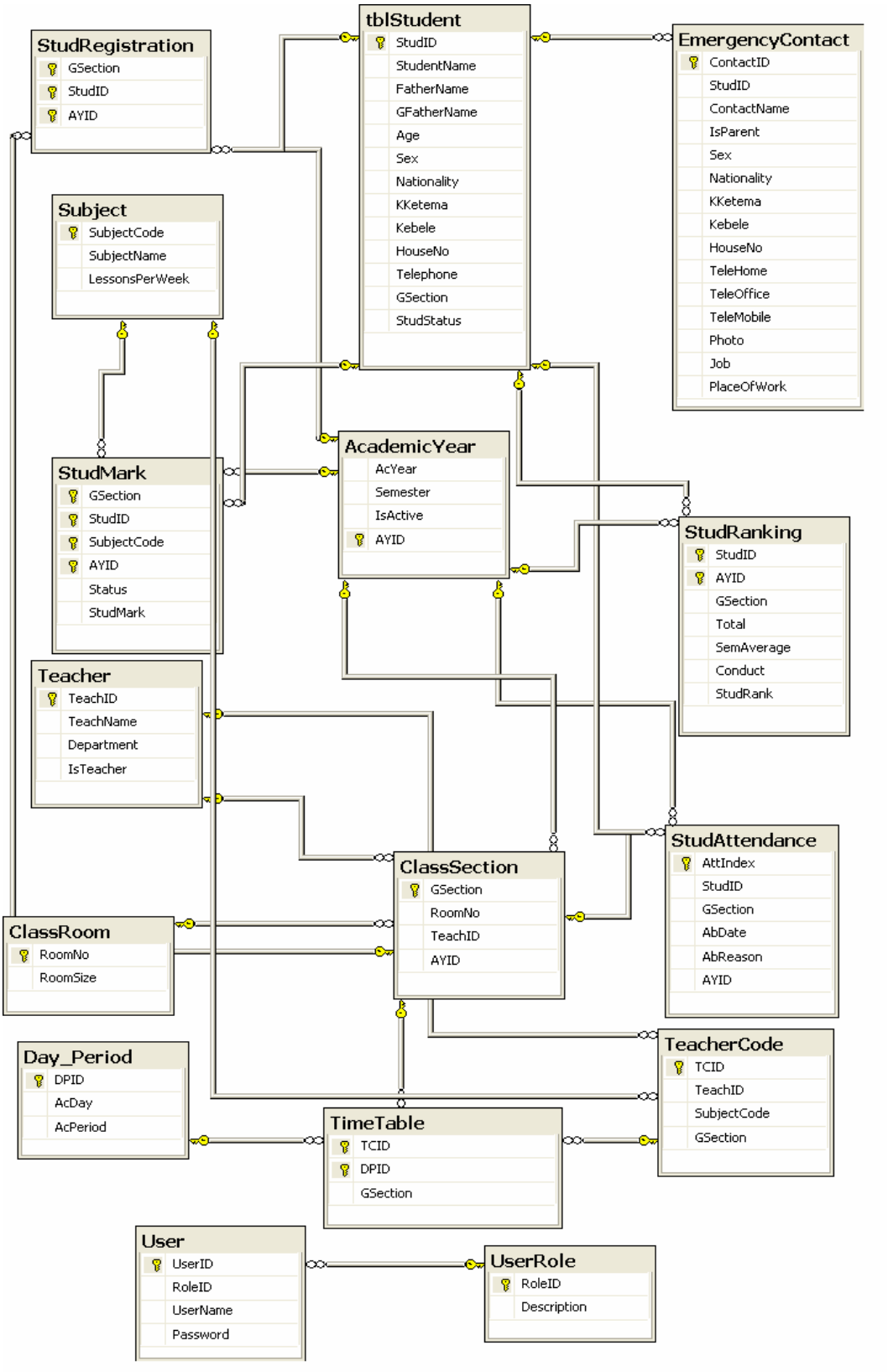


Figure 5.6 Relationship Diagram of the Tables

5.6 Algorithmic Design for the Timetable

The algorithm tries to assign lessons to periods and a teacher to a particular class at a given time while satisfying a set of constraints in order to produce a feasible timetable. Table 5.1 below shows the number of lessons for each subject per week to be distributed in the time slots of the school days. There are five school days starting from Monday to Friday and in each of these days there are seven periods for each class.

Table 5.1 Number of lessons for each subject

Subjects	No. of lessons per week
Amharic	3
English	6
Mathematics	6
Physics	4
Chemistry	3 (Grade 9)
	4 (Grade 10)
Biology	4 (Grade 9)
	3 (Grade 10)
Geography	2
History	2
Civics	3
HPE	2

A class of students is given a room which is fixed for the class throughout the academic year. To each class, a subject teacher is assigned for each of the ten subjects

Let $C = \{c_1, c_2, \ldots, c_m\}$, $S = \{s_1, s_2, \ldots, s_{10}\}$ and $T = \{t_1, t_2, \ldots, t_n\}$ be the set of all classes, subjects and teachers respectively. A teacher teaches only one type of subject from the ten subjects while a subject can be taught by many teachers. Table 5.2 illustrates some examples of assignment of teachers to subjects and classes.

Table 5.2 An example of the relationship among classes, subjects and teachers

Class	**Subject**	**Teacher**
c_1	s_1	t_1
c_1	s_2	t_2
c_1	s_3	t_5
c_2	s_1	t_1
c_2	s_3	t_6
c_3	s_2	t_3
c_4	s_2	t_4
…	…	…
c_i	s_j, $1 \leq j \leq 10$	t_k

Let D = {d_1, d_2, d_3, d_4, d_5} and P = {p_1, p_2, p_3, …, p_7} be the sets of days and time slots (periods) where |D| = 5 and |P| = 7.

Let M (m_{uv}) be a 2D matrix in which each column v corresponds to a class c where c $\in$ C and each row u corresponds to one combination of (d, p) where d $\in$ D and p $\in$ P. The value of cell m_{uv} is determined as:

- $m_{uv} = s_j t_k$, if this cell is assigned to subject-teacher combination $s_j t_k$.
- m_{uv} = '#', if this cell is available for assignment.

The subject-teacher code $s_j t_k$ uniquely identifies the subject teacher assigned to the class. For example $s_j t_k$ could be E_5, to identify an English teacher or it could be M_3, to identify a Mathematics teacher and so on.

Assigning a teacher to more than one class at the same period leads to violation of one of the hard constraints and hence the timetable will not be feasible. Table 5.3 shows an example of time slot assignment to a lesson.

Table 5.3 An example of time slot assignment

	c_1	c_2	c_3
d_1,p_1	s_1t_1	s_1t_2	#
d_1,p_2	#	#	s_2t_3
d_1,p_3	#	#	#
d_2,p_1	s_4t_5	s_4t_5	#
d_2,p_4	s_1t_1	$s_{10}t_6$	#
d_4,p_7	#	#	#
d_5,p_1	#	#	#
		Not Ok	

The maximum and minimum loads assigned to a teacher are 30 and 6 periods respectively. The *Teacher* and *TeacherCode* tables in the database design as shown in Figure 5.6 contain the following fields to help scheduling:

- Teacher's Name
- Subject Code that the teacher teaches
- Grade level (Grade 9 or Grade 10)
- Class
- Teacher's code

The database consists of other additional tables containing important scheduling information.

The *timetabler* selects a teacher from the *teachers* table and reads the subject that the teacher teaches, number of lessons of the subject, and all the classes which have been assigned to the teacher. The load of the teacher is calculated which can not be greater than the maximum load. The system parses the T*eacher* table row by row until all the teachers are visited or to the end of the rows.

Taking a subject-teacher code of the teacher retrieved, the system selects one of the days (Monday, Tuesday, …, Friday) randomly based on the number of lessons of the subject per week, searches a free slot on the selected day for each class assigned to the teacher, checks if the teacher has been assigned to another class in the same period. If appropriate free slot has been found then

the lesson is assigned to the slot. If appropriate slot has not been found then moving previously assigned lesson to a free slot or swapping two or more lessons has been done. If all the slots in the selected day are filled and swapping couldn’t solve the problem, another day is selected. The process continues until the load of the teacher becomes zero. The algorithmic design of the timetable is shown in Algorithm 5.1.

Algorithm 5.1

Loop until all the teachers in the database are visited
  Select a teacher from the *Teacher* table
    Retrieve the *Subject-Teacher code, Grade-Level, Number of lessons of the subject and all the classes* assigned to the teacher
    Calculate load of the teacher
    If *load* is greater than *maxLoad*
      Display Error Message
        Exit Application
    While load of the teacher not zero
      Select a Day uniquely and randomly from the school days based on the number of lessons of the subject
      For Each Day of the week selected
       For Each class assigned to the teacher
          If *allocatedLesson* of the subject to the class is greater than zero
            If *timeslot* is not '#', *move* to the next *slot*
              If teacher is assigned in the period, *move* to the next *slot*
                If appropriate *slot* is not found *swap* previously assigned classes
                  Assign *lesson* to the *slot*
                  Decrease *allocatedLesson* of the subject
                  Decrease *load* of the teacher
        End For
    End For
  End While
End Loop

Chapter 6

Implementation

In this chapter, the tools used in developing the prototype and the developed system are described.

6.1 Programming Tool

The system has two different applications using the same database. These are the Windows application which is sometimes known as thick-client application and Web application which is known as thin-client application. The Windows application is developed using C#, which is one of the development languages in .NET and is object oriented. The Web application is developed using Active Server Pages (ASP .NET 2.0).

6.2 The SMS Prototype

Here, the implemented system is described. How the user interacts with the system and some of the results of interaction with the system along with the screen shots are described.

When a user starts the application, a login screen is displayed as shown in Figure 6.1 to authenticate the user. If the user has typed the correct user id and password to the login screen, the system displays a splash screen for 3 seconds as shown in Figure 6.2 and then a window containing the main menus of the system as shown in Figure 6.3. The main window displays menus and sub menus based on the role of the user that has logged in.

Figure 6.1 Login Screen

Figure 6.2 Splash Screen

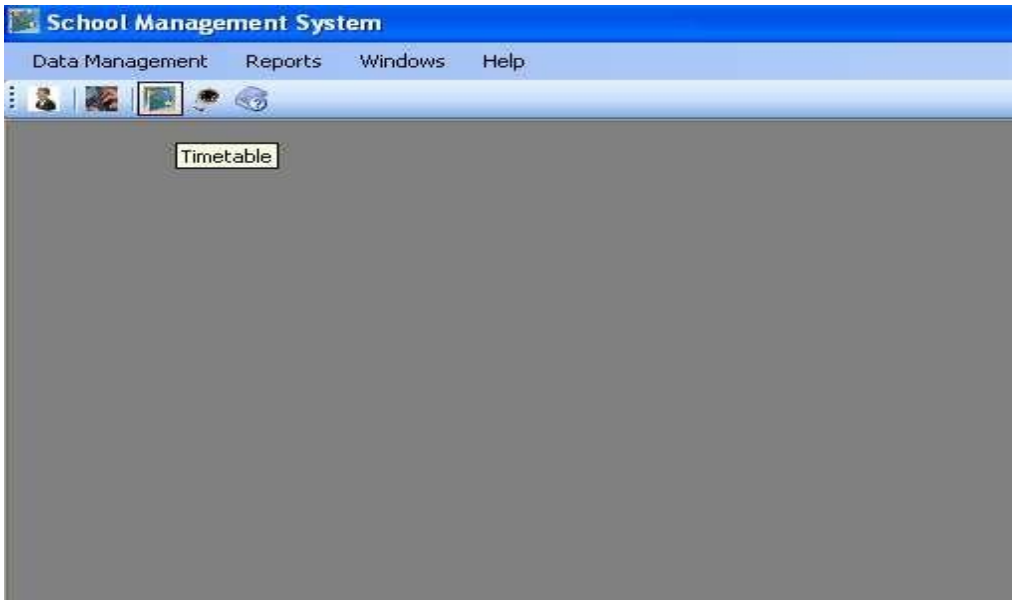
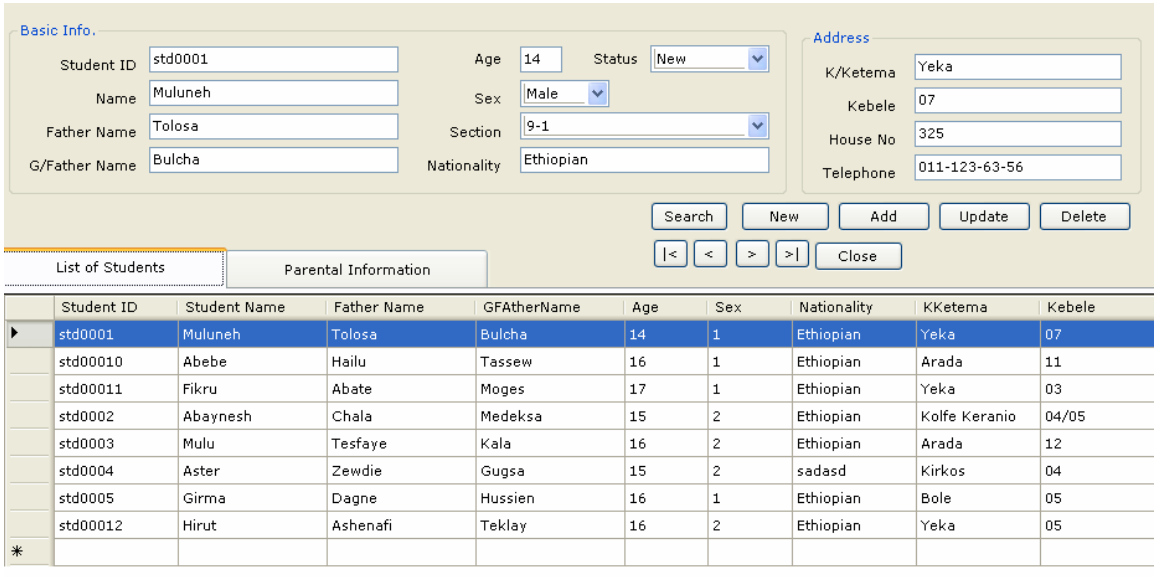


Figure 6.3 Main Menu

The roles for the Windows application are Record Officer, System Administrator and Principal. If the record officer has logged in then the main window displays menus and sub menus to register student along with parental information as shown in Figure 6.4. The system also displays menus and sub menus to record marks, generate report cards and Transcripts.



Student ID	Student Name	Father Name	GFAtherName	Age	Sex	Nationality	KKetema	Kebele
std0001	Muluneh	Tolosa	Bulcha	14	1	Ethiopian	Yeka	07
std00010	Abebe	Hailu	Tassew	16	1	Ethiopian	Arada	11
std00011	Fikru	Abate	Moges	17	1	Ethiopian	Yeka	03
std0002	Abaynesh	Chala	Medeksa	15	2	Ethiopian	Kolfe Keranio	04/05
std0003	Mulu	Tesfaye	Kala	16	2	Ethiopian	Arada	12
std0004	Aster	Zewdie	Gugsa	15	2	sadasd	Kirkos	04
std0005	Girma	Dagne	Hussien	16	1	Ethiopian	Bole	05
std00012	Hirut	Ashenafi	Teklay	16	2	Ethiopian	Yeka	05

Figure 6.4 Screen shot of student registration form

After the student has been registered, the system facilitates to register the student's parent and emergency persons by providing another window as shown in figure 6.5. Note that parental and emergency person information is highly required by the school for handling the student in his/her stay at the school, for example in the case of taking disciplinary measures for the case that the student has committed some crimes. So such information as the name, address of the parent or emergency person can be easily retrieved.

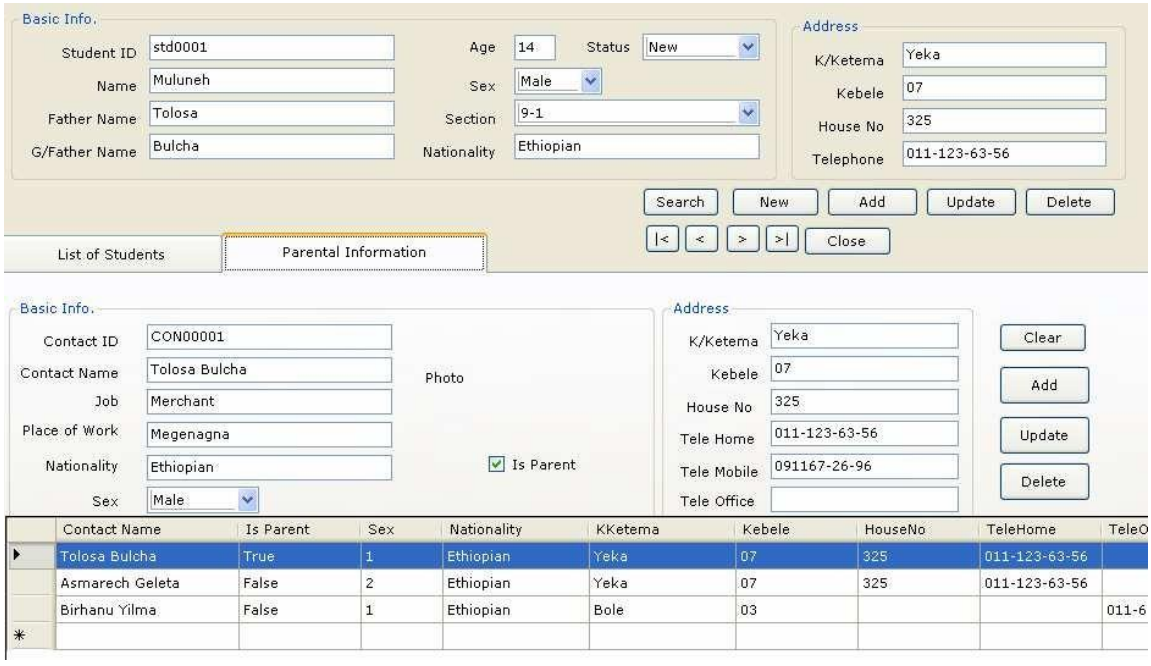
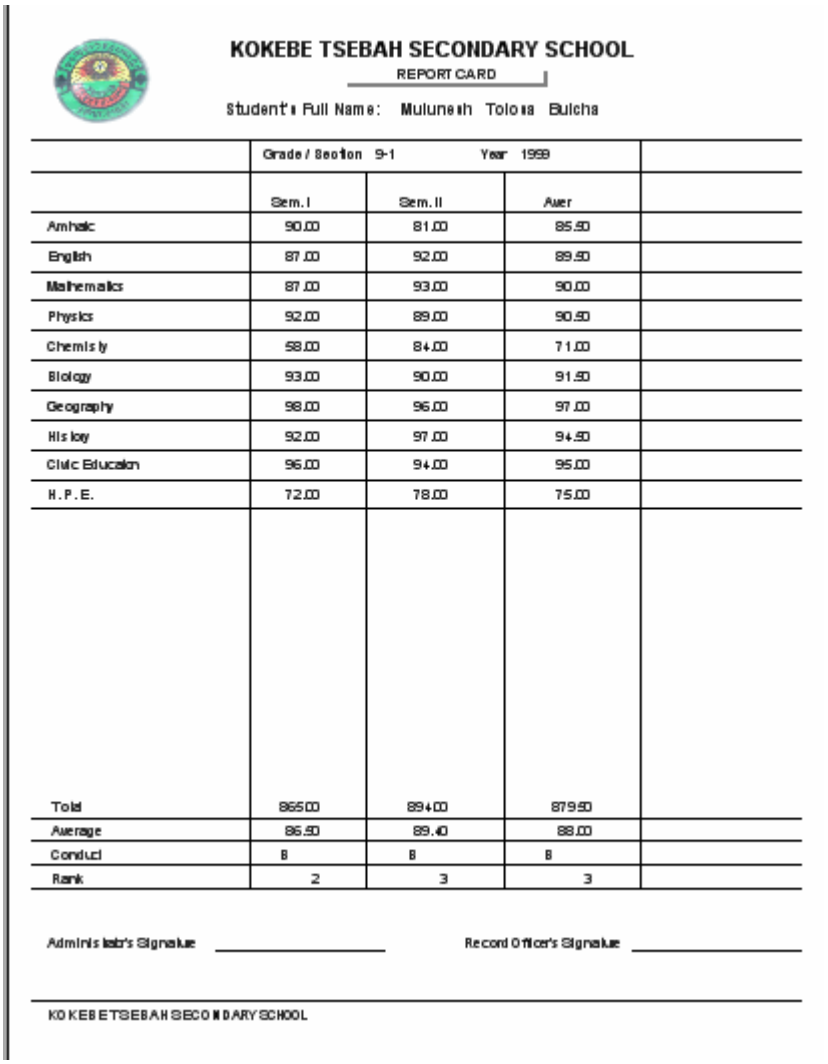


Figure 6.5 Parental and Emergency Person Data Entry and Retrieval Form

The other role of the record officer is to record marks and generate transcript and report cards. These are routine tasks which take much of the times of the user, but once the marks are recorded the tasks are simplified. When the user wants to generate report cards or transcripts, he/she selects the corresponding sub menu from the "Reports" menu and a form is displayed to facilitate generation of the reports. A screen shot of a report card is shown in Figure 6.6 and the transcript report has a similar format.



KOKEBE TSEBAH SECONDARY SCHOOL

REPORT CARD

Student's Full Name: Mulunesh Tolosa Bulcha

	Grade / Section 9-1		Year 1999	
	Sem.I	Sem.II	Aver	
Amharic	90.00	81.00	85.50	
English	87.00	92.00	89.50	
Mathematics	87.00	93.00	90.00	
Physics	92.00	89.00	90.50	
Chemistry	58.00	84.00	71.00	
Biology	93.00	90.00	91.50	
Geography	98.00	96.00	97.00	
History	92.00	97.00	94.50	
Civic Education	96.00	94.00	95.00	
H.P.E.	72.00	78.00	75.00	
Total	865.00	894.00	879.50	
Average	86.50	89.40	88.00	
Conduct	B	B	B	
Rank	2	3	3	

Administrator's Signature _____ Record Officer's Signature _____

KOKEBE TSEBAH SECONDARY SCHOOL

Figure 6.6 Screen shot of Report Card

If the user has logged in as a principal to produce timetable then a form as shown in Figure 6.7 is displayed. When the user clicks on "Generate Timetable" button the scheduler selects a subject-teacher from the database, retrieves all the classes assigned to the teacher, calculates the load of the teacher which cannot be greater than the maximum load and selects one of the days randomly based on the number of lessons of the subject, searches a free appropriate time slot and assigns the slot to the lesson. The scheduler repeats the process until the load of the teacher becomes zero and all the teachers in the database are visited.

The system produces a feasible timetable satisfying the constraints as shown in Figure 6.7. Due to screen size we have limited the table to be displayed for print into a page per class/section as shown in figure 6.8 which could be given to each class. The system also facilitates a timetable to be generated and printed for each teacher as shown in figure 6.9.

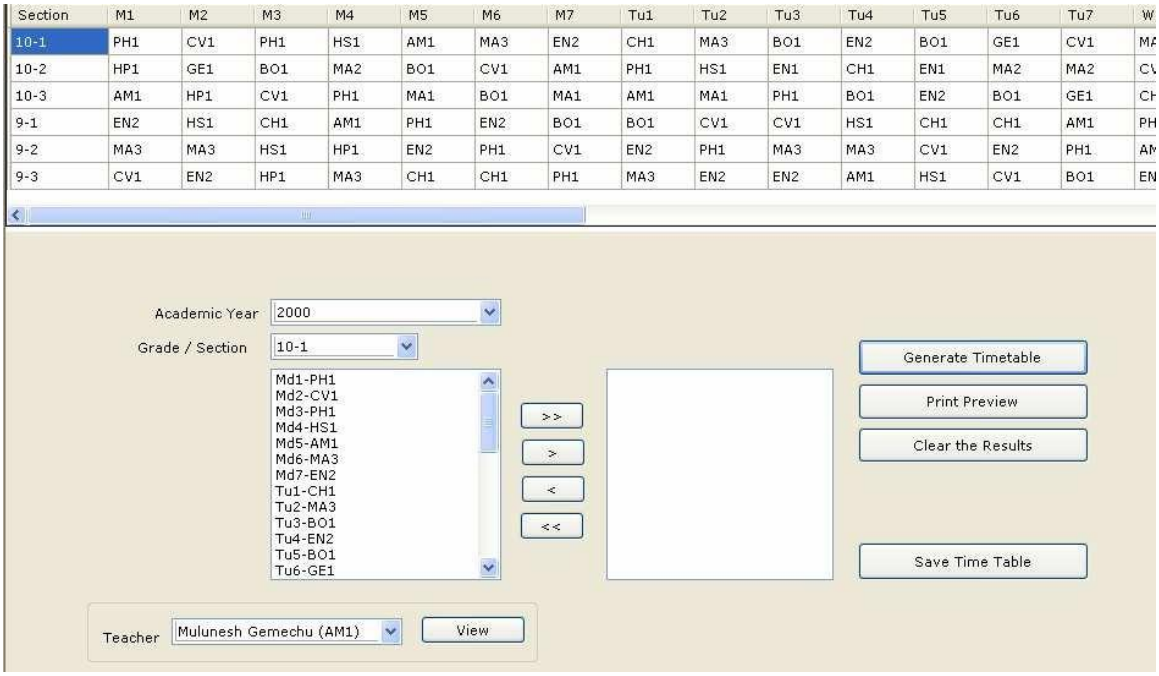


Figure 6.7 Screen shot that shows automatic generation of the timetable

Grade / Section 10-1

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7
Monday	EN2	PH1	CV2	PH1	HS1	AM1	MA3
Tuesday	EN2	CH2	MA3	BO2	BO2	EN2	GE2
Wednesday	CV2	MA3	CV2	MA3	AM1	EN2	AM1
Thursday	MA3	PH1	HP1	PH1	HS1	EN2	CH2
Friday	HP1	CH2	EN2	CH2	MA3	BO2	GE2

Print

Figure 6.8 Timetable for class/section 10-2 Students.

Teacher Code AM1

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7
Monday	9-1	--	--	9-2	--	10-1	10-2
Tuesday	9-1	--	--	--	--	--	9-2
Wednesday	9-1	10-2	10-2	--	10-1	--	10-1
Thursday	--	--	--	--	9-2	--	--
Friday	--	--	--	--	--	--	--

Print

Figure 6.9 Timetable for a teacher (an Amharic teacher Code AM1).

The web application is deployed on the central server in the schools intranet and the clients in the intranet with any browser can access the (home page) school's site. The user logs in by providing his/her user Id and password and then the system displays information based on the role of the logged in user. The roles are homeroom teacher to record attendance, parent and officials to see the status of students and view reports. When the homeroom teacher logged in to record attendance the form shown in figure 6.10 is displayed. This helps the teacher to record absentees and to view the number of dates that a student has been absent. When a student logged in to view his/her status or parent logged in to view the status of his/her child the form shown in figure 6.11 is displayed.

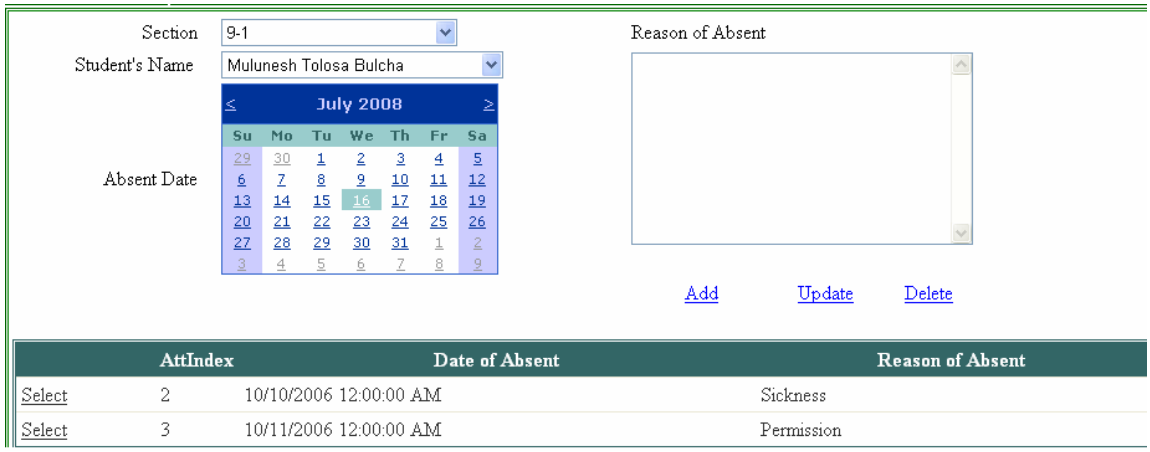


Figure 6.10 Screen shot of attendance recording page

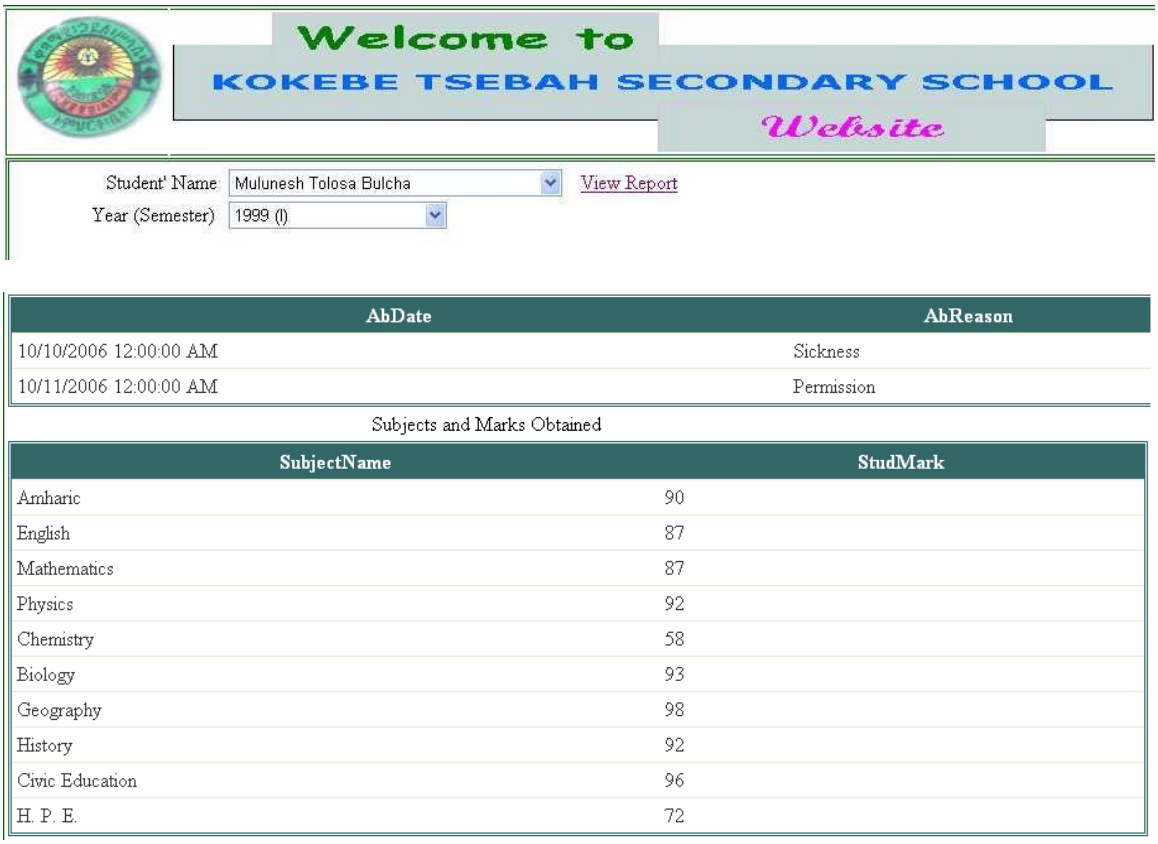


Figure 6.11 Screen shot showing the page displayed for a parent

Chapter 7

Conclusion and Recommendations

7.1 Conclusion

In this project, we developed an automated school management system that facilitates the various activities taking place at schools.

The system developed in the project consists of windows and web applications. These are two different applications on the same database. The windows application takes most of the activities such as offline student registering, transcript and report card generation and producing the timetable. The web application facilitates attendance recording by the homeroom teachers and to view reports, to view status of students by students, teachers and parents.

Our solution of the timetabling problem is very simple. Data structures are used to implement the timetable designed. The scheduler selects a subject-teacher from the database, retrieves all the classes assigned to the teacher, calculates the load of the teacher which cannot be greater than the maximum load and selects one of the days randomly based on the number of lessons of the subject, searches a free appropriate time slot and assigns the slot to the lesson. The scheduler repeats the process until the load of the teacher becomes zero and all the teachers in the database are visited. Finally the result generated is stored in a database.

The prototype has been tested with data from Kokebe Tsebah Secondary School. It has been shown that the system effectively registers students along with parental information, easily retrieves information about a student and generates the required reports such as transcript, report card and timetable. In addition to generating a feasible master timetable it produces a timetable for each teacher. Further more it has been shown that the web application of the system helps attendance recording by the homeroom teacher and parents can view the status of their children using the Internet or Intranet of the school.

7.2 Recommendations

To enhance the efficiency of the system, in the following we have listed some recommendations and future works.

As education is central to development there should be a good facility to make stakeholders participate in school improvement programs and decision making. Parents and Education Bureaus from Kebele and Kifle-ketema are among the stake holders. To facilitate easy information access to such bodies the web application could be further enhanced by incorporating additional reports required by Kebele and Kifle-ketema Education Bureaus. Such facilities will increase participants in decision making at educational activities and students achievement.

We also believe that timetables should be flexible. In real world situations there are preferences. A restriction of the sort that every teacher should have some specific free periods or some part of days off requires an efficient search technique. Efficiency of the timetable could be further enhanced by improving the search technique so that such constraints as preferences could be taken into consideration.

References

[1]. E. Burke and W. Erben. Practice and Theory of Automated Timetabling, Third International Conference, Germany, Springer Private Limited, August 2000

[2]. J. G. Hedberg et. al. (1992). Educational information systems: Problems of the small educational organisation. Australian Journal of Educational Technology, 8(2), 132-160. http://www.ascilite.org.au/ajet/ajet8/hedberg.html

[3]. M. Marte. Models and Algorithms for School Timetabling, A Constraint-Programming Approach, Ph.D dissertation, an der Fakult¨at f¨ur Mathematik, Informatik und Statistik der Ludwig-Maximilians-Universit ¨at M¨unchen, July, 2002

[4] R.J. Willemen. School Timetable Construction: Algorithms and Complexity, Thesis, Faculty of Mathematics and Computer Science, Technische Universiteit Eindhoven, 2002.

[5]. S. Petrovic and E. Burke. University Timetabling, School of Computer Science and Information Technology, University of Nottingham, 2002, pp. 1-4

[6] T. Willis and B. Newsome. Beginning Visual Basic 2005, Wiley Publishing, Inc., 2006.